AA 2022/2023

Machine Learning for Modelling: *Supervised Learning*

Simone Bianco

1

AA 2022/2023

SIFT – Scale Invariant Feature Transform

2

Object Recognition

Classes of objects

Class 1: cars

Class 2: animals

3

Object Recognition

Object instances

Instance 1: Tesla Model S

Instance 2: Book «The Martian» by Andy Weir

4

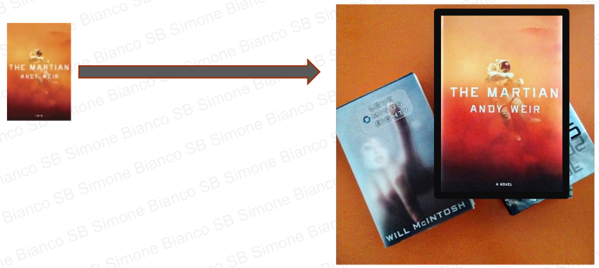
AA 2022/2023

Object instance recognition

5

Task

- Detect if a particular object is present inside the image
- Localize (precisely) the object



6

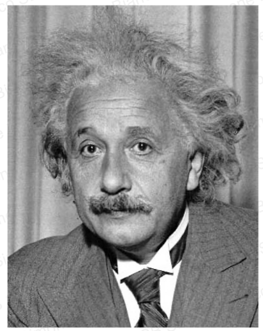
Template matching

- Intuitive approach
- Computationally inefficient: we have to try all possible combinations of locations, scale, rotations, etc.

7

Template matching

- **Goal**: find  in the image
- Main challenge: What is a good similarity or distance measure between two patches?



8

Matching with filters

Goal: find in image

Method: SSD (*Sum of Squared Differences*)

$$h[m,n]=\sum_{k,l}(g[k,l]-f[m+k,n+l])^2$$

Input 1- sqrt(SSD) Thresholded Image

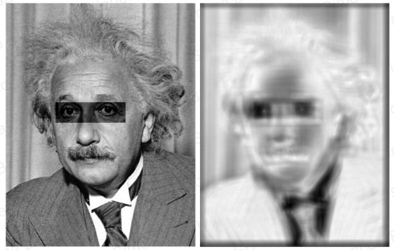
9

Matching with filters

Goal: find in image

Method: SSD (*Sum of Squared Differences*)

$$h[m,n]=\sum_{k,l}(g[k,l]-f[m+k,n+l])^2$$



Input 1- sqrt(SSD)

10

Matching with filters

Goal: find in image

Method : Normalized cross-correlation

mean template

mean image patch

$$h[m,n]=\frac{\sum_{k,l}(g[k,l]-\bar{g})(f[m-k,n-l]-\bar{f}_{m,n})}{\left(\sum_{k,l}(g[k,l]-\bar{g})^2\sum_{k,l}(f[m-k,n-l]-\bar{f}_{m,n})^2\right)^{0.5}}$$

11

Matching with filters

Goal: find in image

Method : Normalized cross-correlation

Input Normalized X-Correlation Thresholded Image

12

Matching with filters

Goal: find [eye] in image

Method : Normalized cross-correlation

Input | Normalized X-Correlation | Thresholded Image

13

Matching with filters

Q: What is the best method to use?

SSD

- Faster
- Sensitive to overall intensity

Normalized cross-correlation

- Slower
- Invariant to local average intensity
- Invariant to local contrast

14

Template matching – Sliding window

Example

A chair | Find the chair in this image | Output template matching with pixel difference

15

Template matching – Sliding window

Example

A chair | Find the chair in this image | Output template matching with pixel difference

Pretty much garbage

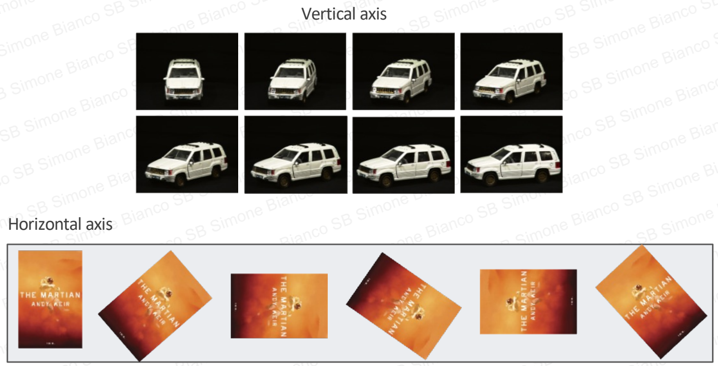
Single template matching is not going to make it

16

Variability – Scale, Translation

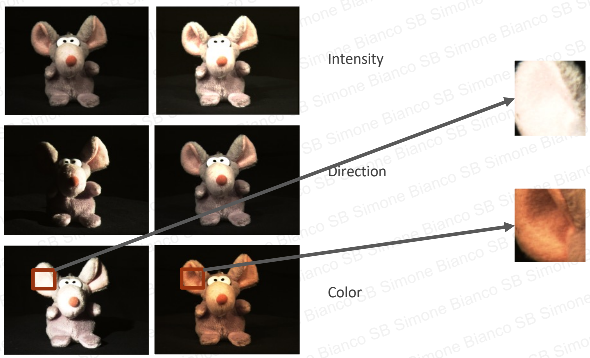
17

Variability - Rotation



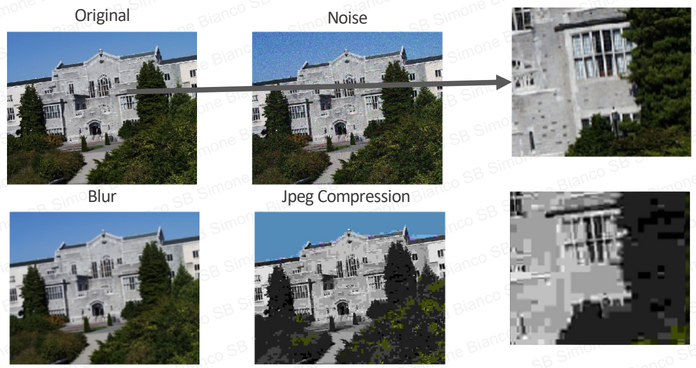
18

Variability - illumination



19

Variability – Image quality



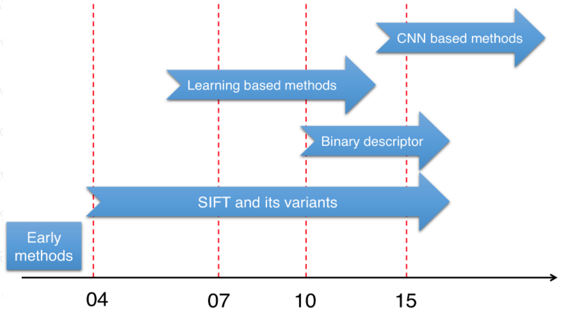
20

AA 2022/2023

Local descriptors trends

21

Local Descriptors Trends



22

SIFT and SURF : classic local features

SIFT: Scale Invariant Feature Transform (Lowe '99 and '04)

- The algorithm is patented in Canada by the University of British Columbia
- published by David Lowe in 1999

SURF: Speeded Up Robust Features (Bay '04)

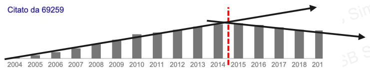
- 3-7 times faster than SIFT, with similar matching performance.
- More than 8000 citations.
- CVIU Most Cited Paper
- Koenderink Prize of ECCV'16
- Included in OpenCV

23

SIFT Paper Citation trends

Distinctive image features from scale-invariant keypoints

Autori	David G Lowe
Data pubblicazione	2004/11/1
Pubblicazione	International journal of computer vision
Volume	60
Numero	2
Pagine	91-110
Editore	Springer Netherlands
Descrizione	This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images. This paper also describes an approach to using these features for object recognition. The recognition proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object, and finally performing verification through …
Citazioni totali	Citato da 69259



Articoli in Scholar	Distinctive image features from scale-invariant keypoints DG Lowe - International journal of computer vision, 2004 Citato da 69259 Articoli correlati Tutte e 148 le versioni

- SIFT paper is among the most cited computer vision papers ever.

- But it's not as dominant as it once was.

24

Why Keypoints-based methods?

Keypoints Remain Relevant

- When accurate geometric recovery matters, they remain unequaled.
- They are efficient for real-time applications.
- They provide an effective way
  - to compress the information present in large images,
  - to recognize specific locations.
- The algorithms do not need to be retrained for each new application.

Future algorithms will combine Deep Learning and keypoint matching.

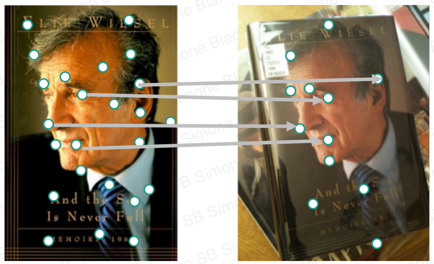
25

AA 2022/2023

Keypoints-Based Approaches

26

Keypoints-based approaches



Steps
- Keypoint detection
- Keypoint description
- Matching of similar keypoints
- Similarity score based on matching points

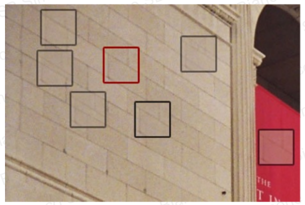
- Use of Interest points (keypoints)
- Do not need to try all the combinations of transformations.

27

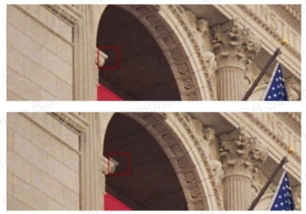
Harris Corner Detector

- Corners are better than edges!
- How to define a corner?
- Harris at al. [1] ➡ Patches that generate a large variation when moved around

Example of a **bad** corner



Example of a **good** corner



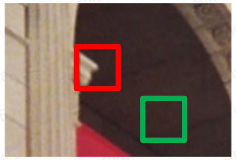
[1] Harris, Chris, and Mike Stephens. "A combined corner and edge detector." *Alvey vision conference*. Vol. 15. 1988.

28

Harris Corner Detector

Patches that generate a large variation when moved around

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2$$



- E is the difference between the original and the moved window.
- u is the window's displacement in the x direction
- v is the window's displacement in the y direction
- w(x, y) is the window at position (x, y). This acts like a mask. Ensuring that only the desired window is used.
- I is the intensity of the image at a position (x, y)
- I(x+u, y+v) is the intensity of the moved window
- I(x, y) is the intensity of the original window

[1] Harris, Chris, and Mike Stephens. "A combined corner and edge detector." *Alvey vision conference*. Vol. 15. 1988.

29

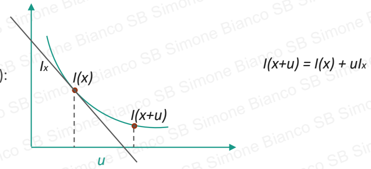
Harris Corner Detector

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2$$ ← Patches that generate a large variation when moved around

$$\sum_{x,y}[I(x+u, y+v) - I(x,y)]^2$$ ← Leave out the weights (for simplification)

$$E(u,v) \approx \sum_{x,y}[\boxed{I(x,y) + uI_x + vI_y} - I(x,y)]^2$$ ← Can be approximated with the first Taylor expansion

Math hints (derivative):



30

Harris Corner Detector

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2$$ ← Patches that generate a large variation when moved around

$$\sum_{x,y}[I(x+u, y+v) - I(x,y)]^2$$ ← Leave out the weights (for simplification)

$$E(u,v) \approx \sum_{x,y}[I(x,y) + uI_x + vI_y - I(x,y)]^2$$ ← Can be approximated with the first Taylor expansion

$$E(u,v) \approx \sum_{x,y} u^2 I_x{}^2 + 2uvI_xI_y + v^2 I_y{}^2$$ ← Square expansion

$$E(u,v) \approx [u \quad v]\left(\sum \begin{bmatrix} I_x{}^2 & I_xI_y \\ I_xI_y & I_y{}^2 \end{bmatrix}\right)\begin{bmatrix} u \\ v \end{bmatrix}$$ ← Conversion into a matrix

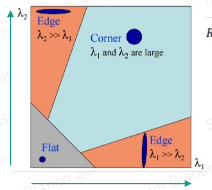
$$M = \sum w(x,y)\begin{bmatrix} I_x{}^2 & I_xI_y \\ I_xI_y & I_y{}^2 \end{bmatrix} \qquad E(u,v) \approx [u \quad v]\, M \begin{bmatrix} u \\ v \end{bmatrix}$$ ← Another way to write it

31

Harris Corner Detector

$$M = \sum w(x,y)\begin{bmatrix} I_x{}^2 & I_xI_y \\ I_xI_y & I_y{}^2 \end{bmatrix}$$ ← M matrix (see slides before)

Eigenvalues of the matrix can help determine the suitability of a window:



$$R = \det M - k(trace\ M)^2$$ ← All windows that have a **score R** greater than a certain value are corners.

$$\det M = \lambda_1\lambda_2$$ ← $\lambda_1\lambda_2$ are the eigenvalues

$$trace\ M = \lambda_1 + \lambda_2$$

[1] Images credits: http://aishack.in/tutorials/harris-corner-detector/ very good website with lot of didactic material

32

SIFT

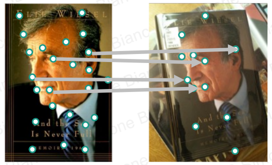
- SIFT [1] (Scale Invariant Features Transform)
- Author: David Lowe. University of British Columbia –Canada

Main difference with Harris detector:

- Not only a Keypoint Detector but also a **Keypoint Descriptor**

Robust to:

- Scale
- Rotation
- Illumination
- Viewpoint



[1] David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.

33

Kepoint detection – Processing steps

Keypoint detection

Construction of the Scale Space
↓
Find maxima/minima
↓
Discard bad points
↓
Compute orientations

↓

Keypoint description

34

Scale Space

Increasing blur
Increasing scale

- Needed to obtain **Scale Invariance**
- Higher scales = higher blur
- Higher blur = less details

Blurred image
Gaussian blur operator
Original Image

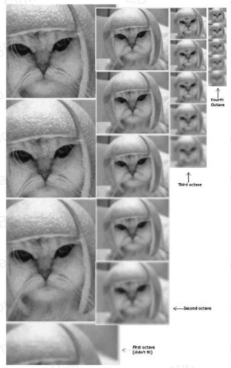
$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$

Scale parameter: higher value higher blur

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

35

Scale Space - Octaves

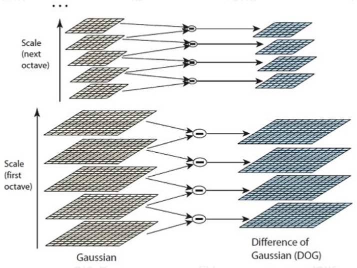


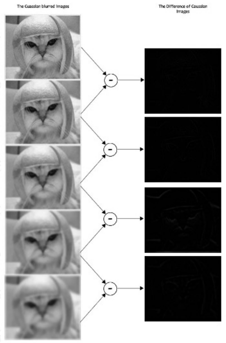
- Scales and **Octaves**
- Usually 4 octaves and 5 blur levels
- Reduces computations

36

Difference Of Gaussians

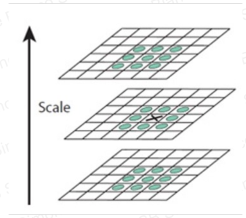
- Used to detect edges and corners
- Laplacian of Gaussian (LoG) approximation
- Very fast and efficient (only subtraction)





37

Locate maxima/minima in DoG images



- X is the current pixel
- Green circles are the considered neighbours
- 26 (=9+9+8) checks.
- X is a **keypoint** if it is the greatest of all its neighbours
- Usually there is no need to check all the neighbours. Discard happens after few checks.

39

Refinements

To be removed
- Low Contrast Points
  - Check for the value of pixel in the DoG image
  - If value is under threshold the point is rejected
- Edges
  - Compute gradients in the two directions.
  - Gradient in one direction is just a difference of pixel values

Flat region

40	40	40
41	45	42
40	41	40

Abs difference
41 – 40 = 1
42 – 41 = 1

↕ Small ↔ Small

Edge

20	20	20
40	45	40
40	40	40

↕ Big ↔ Small

Corner

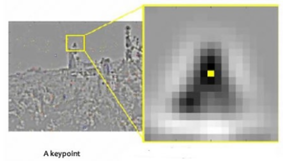
20	20	20
20	45	40
20	40	40

↕ Big ↔ Big

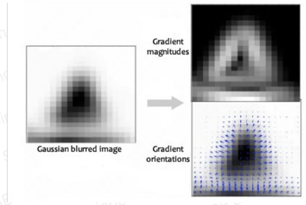
Searching for this!

40

Keypoint Orientation



- Goal: achieve **rotation invariance**
- Histogram of Orientations
- Weighted by Gradient magnitudes



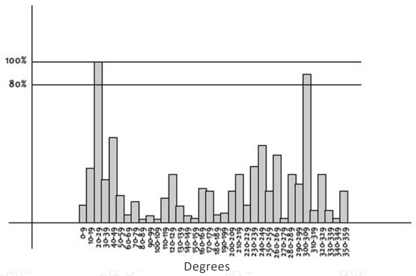
Gradient magnitude — Gradient orientation — Gaussian blurred image

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

41

Keypoint Orientation - Histogram



- 36 bin (10 degrees per bin)
- Dominant orientation given by the highest peak
- Peaks higher than the 80% of the highest are kept (the keypoint is split).
- New keypoint has the same scale and location but different orientation

42

Keypoint description – Processing steps

Keypoint detection

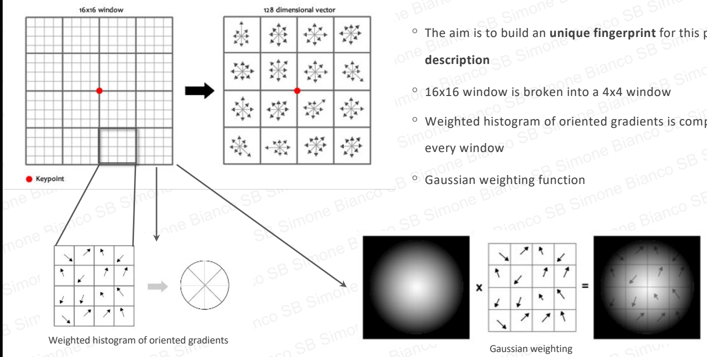
↓

Keypoint description:

- Split the window in 4x4 sub-windows
- ↓ Compute weighted histograms
- ↓ Gaussian weighting
- ↓ L2 normalization
- ↓ Subtract keypoint orientation
- ↓ Cut off values > 0.2 and renormalize

43

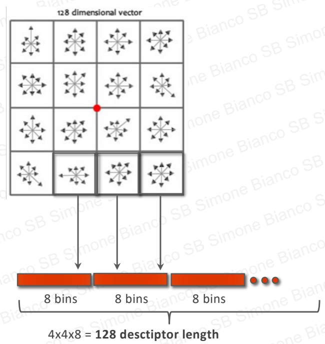
Keypoint Description – part 1



- The aim is to build an **unique fingerprint** for this point: a **description**
- 16x16 window is broken into a 4x4 window
- Weighted histogram of oriented gradients is computed for every window
- Gaussian weighting function

44

Keypoint Description – part 2



- Concatenation of the 8 bins of each of the 4x4 sub-window.
- L2 Normalization (divide by root sum of squares)

Rotation invariance:

The keypoint orientation is subtracted to every bin orientation

Illumination invariance:

Every value over 0.2 is set to 0.2. The vector is then renormalized

[1] Images credits: http://aishack.in/tutorials/sift-scale-invariant-feature-transform-introduction/ very good website with didactic material

45

SIFT - Summary

Keypoint detection
- Construction of the Scale Space
- ↓ Difference of Gaussians
- ↓ Find maxima/minima
- ↓ Discard edges and low-contrast points
- ↓ Compute orientations

Keypoint description
- Split the window in 4x4 sub-windows
- ↓ Compute weighted histograms
- ↓ Gaussian weighting
- ↓ L2 normalization
- ↓ Subtract keypoint orientation
- ↓ Cut off values > 0.2 and renormalize

46

List of keypoint detectors/descriptors

SIFT implementations

Detector	Descriptor	Dimensionality
SIFT Lowe (SIFT 1999)	SIFT Lowe	128
SIFT OpenCV	SIFT OpenCV	128
SIFT VLFeat (Vedaldi 2010)	SIFT VLFeat	128

Affine Invariant Detectors

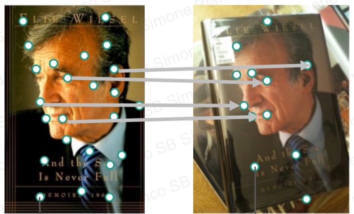
Detector	Descriptor	Dimensionality
DoG	SIFT	128
Multiscale-Harris	SIFT	128
Harris-Laplace	SIFT	128
Hessian	SIFT	128
Multiscale-Hessian	SIFT	128
Hessian-Laplace	SIFT	128

Others

Detector	Descriptor	Dimensionality
SURF	SURF (Bay 2006)	64
SURF	FREAK (Alahi 2012)	64
Kaze	Kaze (Alcantarilla 2012)	64

47

Keypoint matching



x (128 dims vector)

y (128 dims vector)

d(**x**,**y**)

How do I match similar points?

↓

Euclidean distance

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\ldots+(x_n-y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$$

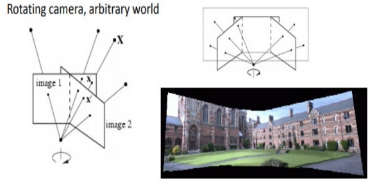
48

Homography

Briefly, the planar homography relates the transformation between two planes

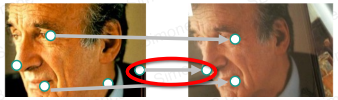
$$s\begin{bmatrix}x'\\y'\\1\end{bmatrix} = H\begin{bmatrix}x\\y\\1\end{bmatrix} = \begin{bmatrix}h_{11}&h_{12}&h_{13}\\h_{21}&h_{22}&h_{23}\\h_{31}&h_{32}&h_{33}\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix}$$

Rotating camera, arbitrary world



49

RANSAC



RANSAC – RANdom SAmple Consensus

Used to discard false matches

Algorithm outline:

1. Select at least four feature pairs (at random)
2. Compute homography H (exact)
3. Compute inliers where $SSD(p', Hp) < \varepsilon$
4. Keep largest set of inliers
5. Re-compute least-squares $\boldsymbol{H}$ estimate on all of the inliers

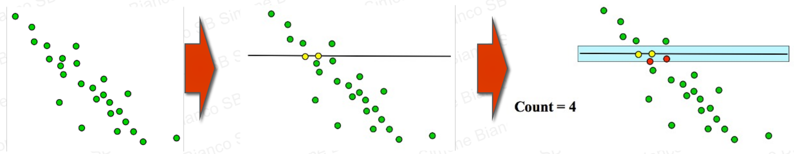
Iterate multiple times

50

RANSAC

First iteration

Points set | Random points selection | Count number of inlayer points

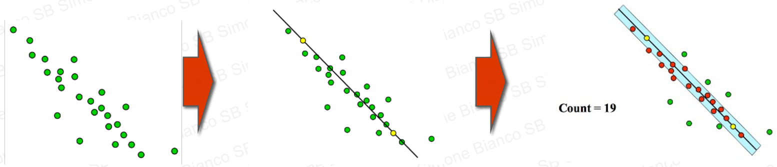


51

RANSAC

Second iteration

Points set | Random points selection | Count number of inlayer points

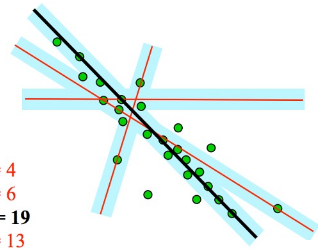


52

RANSAC

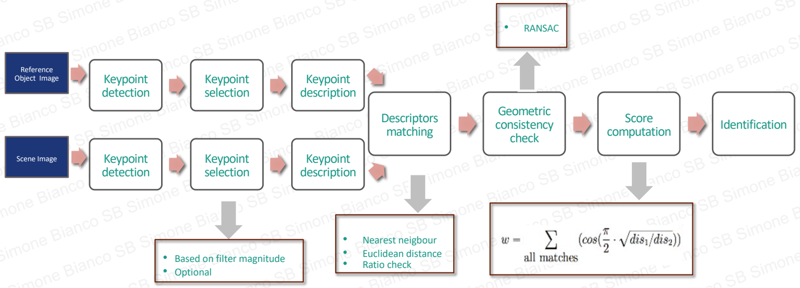
- Choose the model with the highest number of inlayers.
- Simple example with lines.
- The same algorithm is used with more complex models (e.g. to handle affine transformations).
- Robust even if the majority of points are outlayers

Count = 4
Count = 6
Count = 19
Count = 13



53

Overview of the complete pipeline



54

Identification

Example formula used to check the correspondence:

$$w = \sum_{\text{all matches}} (cos(\frac{\pi}{2} \cdot \sqrt{dis_1/dis_2}))$$

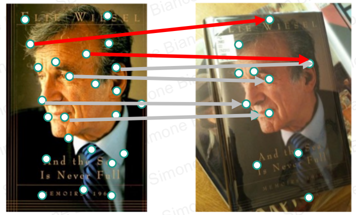
All correct matches

Distance from the nearest and second nearest.

To verify how strong the matching is.

Final score

- To be thresholded
- If w is over the threshold the two images contain the same object



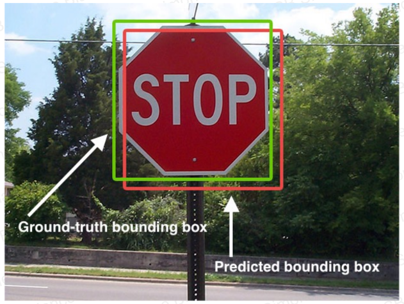
55

AA 2022/2023

Object detection: evaluation metrics

56

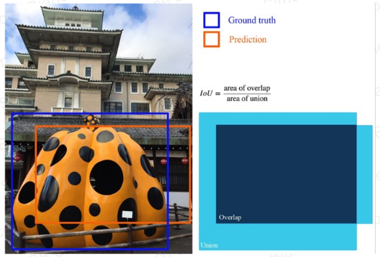
Intersection Over Union



57

Intersection Over Union

IoU = $\frac{\text{Area of Overlap}}{\text{Area of Union}}$



58

Intersection Over Union

1

2 Set a threshold (e.g. 0.5)

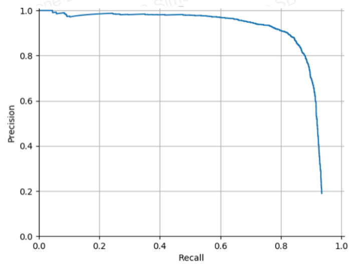
3
TP = True positive
TN = True negative
FP = False positive
FN = False negative

4
$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN}$$

59

mean Average Precision curve

By varying the threshold we obtain the mAP curve



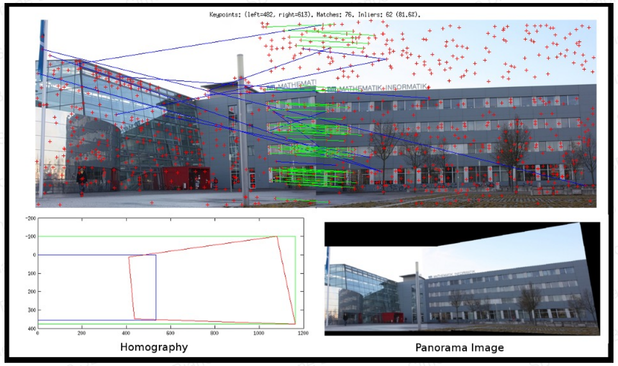
60

AA 2022/2023

Applications

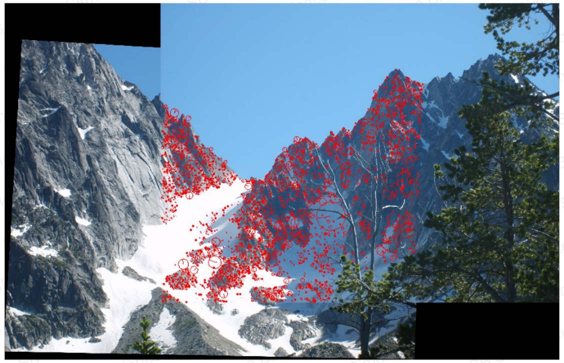
61

Applications – Panorama Stitching



62

Applications – Panorama Stitching

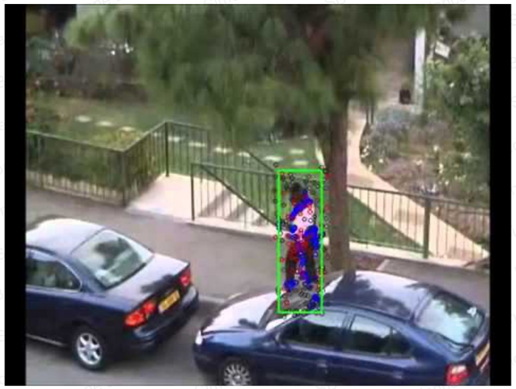


63

Applications – Video Stabilization

64

Applications – Tracking



65

Applications – Augmented reality - IKEA

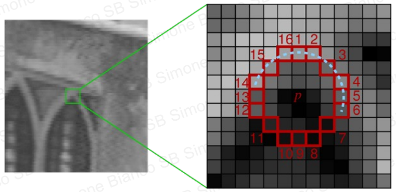
66

AA 2022/2023

Variation on the theme

67

FAST – another keypoint detector



- **FAST – Features from accelerated segment test**
- For every pixel check the value of the neighbors
- If the value of at least N neighbors is less then the value of the center pixel, this is a corner

Tips for faster computation:

- First compare values of pixels 1,5,9,13
- At least 3 of these pixel values must be under the value of the central pixel
- If not the keypoint candidate is discarded
- Else everyone of the 16 neighbors is checked

68

Dense SIFT

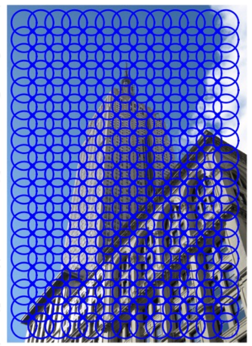
- Instead of computing keypoints use a fixed grid
- Every point in the fixed grid is descripted using the SIFT descriptor

PRO:

- Faster to compute (no keypoint detection)
- Can work with smooth surfaces (few corners)

CONS:

- Corners are usually more distinctive (reliable)
- Slower keypoint comparison (usually a lot more points to be compared)



69

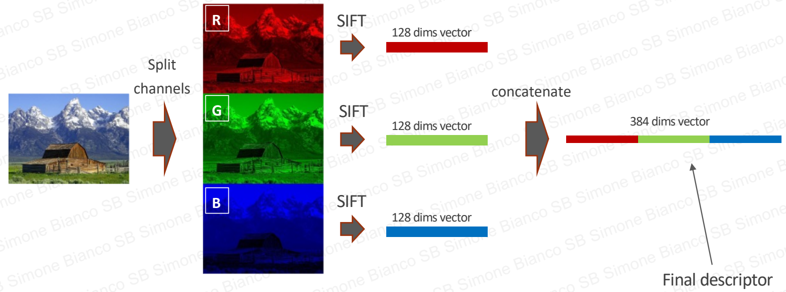
Color descriptors

- SIFT does not use color information
- It just works on grayscale images
- Some objects categories need color information to be distinguished
- Two ways of mixing the color and shape information:
  - Late Fusion approaches
  - Early Fusion approaches

Supermarket Milan Dataset

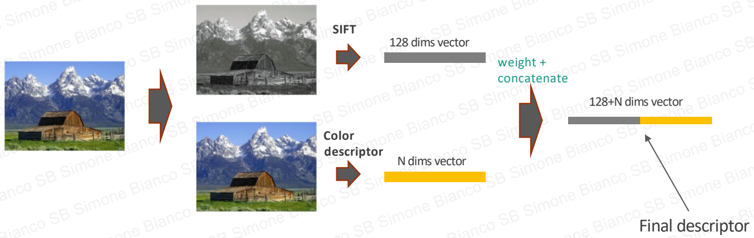
70

Early Fusion approaches



71

Late Fusion approaches



- N depends on the type of descriptor
- Example: simple histogram on color-space
- Before the concatenation of color and shape descriptor usually the two components are weighted
- Possibility to give more importance (higher weight) to color or shape information

72

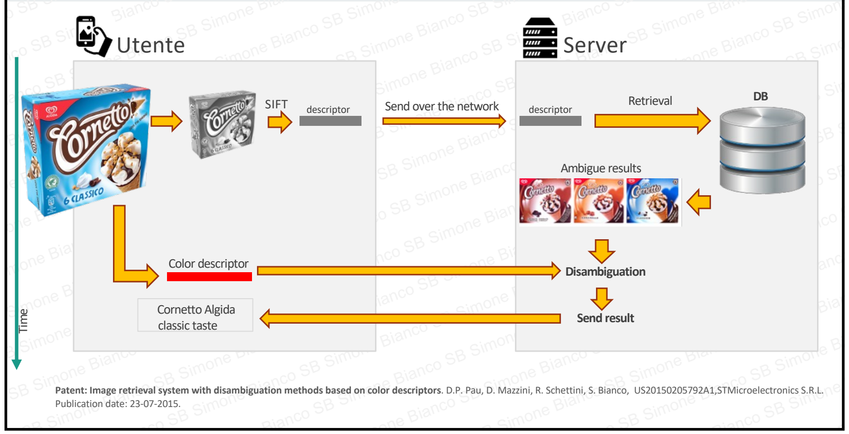
Color Descriptors

Detector	Dimension	Fusion
RGB SIFT [1]	384	Early
Opponent SIFT [1]	384	Early
Transformed Color SIFT [1]	384	Early
HSV SIFT [1]	384	Early
C-SIFT [2]	384	Early
rg-SIFT [1]	256	Early
oRGB-SIFT [1]	384	Early
Hue SIFT [1]	164	Late
Color Names [3]	139	Late
Fuzzy Sets Color Names [3]	139	Late
Discriminative Color [3]	139, 153, 178	Late

[1] K.E. Van De Sande, T. Gevers, C.G. Snoek, **Evaluating color descriptors for object and scene recognition**, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1582–1596.
[2] A.E. Abdel-Hakim, A.A. Farag, **Csift: a sift descriptor with color invariant characteristics**, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE, 2006, pp. 1978–1983.
[3] Van De Weijer, C. Schmid, **Applying color names to image description**, in: IEEE International Conference on Image Processing, 2007, ICIP 2007, vol. 3, IEEE, 2007, p. III-493.

73

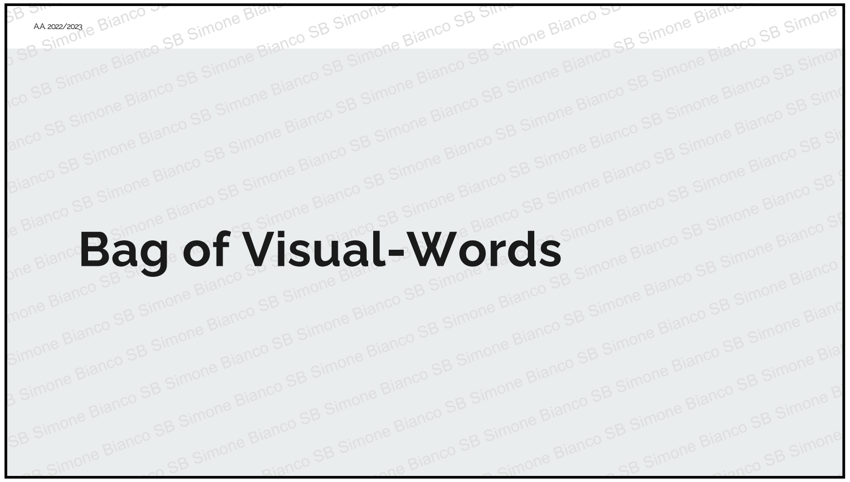
Disambiguation using color descriptors



74

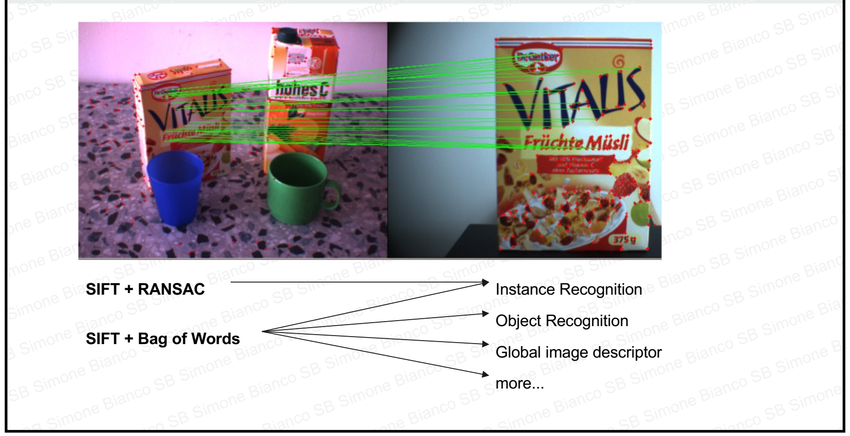
Color descriptors - Applications

75



76

Small Recap



77

Bag-of-words: motivation

President George W. Bush Speech in 2001

ADDRESS TO THE JOINT SESSION OF THE 107TH CONGRESS
UNITED STATES CAPITOL
WASHINGTON, D.C. SEPTEMBER 20, 2001

Mr. Speaker, Mr. President Pro Tempore, members of Congress, and fellow Americans: In the normal course of events, Presidents come to this chamber to report on the state of the Union. Tonight, no such report is needed. It has already been delivered by the American people. We have seen it in the courage of passengers, who rushed terrorists to save others on the ground — passengers like an exceptional man named Todd Beamer. And would you please help me to welcome his wife, Lisa Beamer, here tonight. We have seen the state of our Union in the endurance of rescuers, working past exhaustion. We have seen the unfurling of fl ags, the lighting of candles, the giving of blood, the saying of prayers — in English, Hebrew, and Arabic. We have seen the decency of a loving and giving people who have made the grief of strangers their own. My fellow citizens, for the last nine days, the entire world has seen for itself the state of our Union — and it is strong. Tonight we are a country awakened to danger and called to defend freedom. Our grief has turned to anger, and anger to resolution. Whether we bring our enemies to justice, or bring justice to our enemies, justice will be done. I thank the Congress for its leadership at such an important time. All of America was touched on the evening of the tragedy to see Republicans and Democrats joined together on the steps of this Capitol, singing "God Bless America." And you did more than sing; you acted, by delivering $40 billion to rebuild our communities and meet the needs of our military. Speaker Hastert, Minority Leader Gephardt, Majority Leader Daschle and Senator Lott, I thank you for your friendship, for your leadership and for your service to our country. .. (to be continued)

- We want to represent the <u>**topic**</u> of the document in a <u>**compact**</u> way
- **Orderless** document representation: frequencies of words from a dictionary
  Salton & McGill (1983)

78

Bag-of-words: motivation

US Presidential Speeches Tag Cloud
http://chir.ag/projects/preztags/

79

Bag-of-words: motivation

80

Bag-of-words: motivation

81

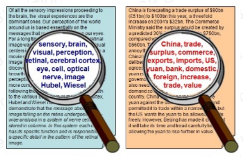
Bag of visual words (Bag of features)

82

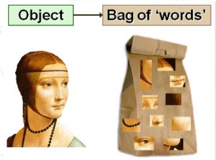
Bag-of-words

- Fei Fei et al. [1]
- From text analysis
- Document representation as a bag of important keywords
- An object is represented as a bag of visual words (patches)
- Allows the recognition of a large collection of objects

Text



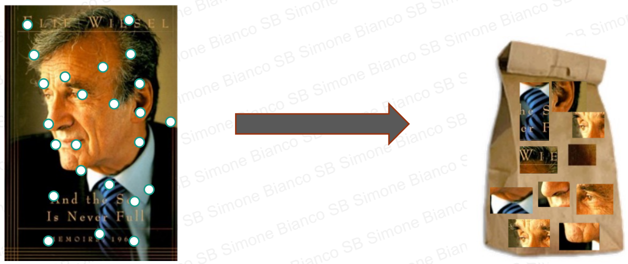
Images



[1] Fei-Fei, Li, and Pietro Perona. "A bayesian hierarchical model for learning natural scene categories." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.* Vol. 2. IEEE, 2005.

83

Bag-of-words



84

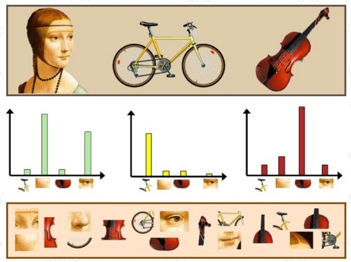
Bag-of-words

a. Extract local features

b. Learn "visual vocabulary"

c. Quantize local features using visual vocabulary

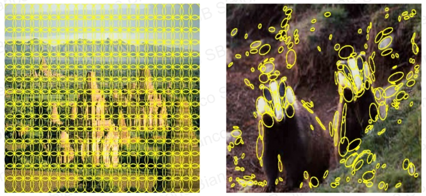
d. Represent images by frequencies of "visual words"



85

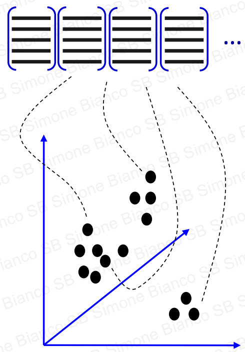
Local features extraction

- Sample patches and extract descriptors



86

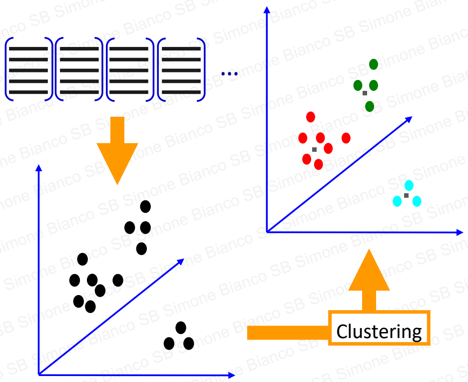
Learning the visual vocabulary (1/2)



Extracted descriptors from the training set

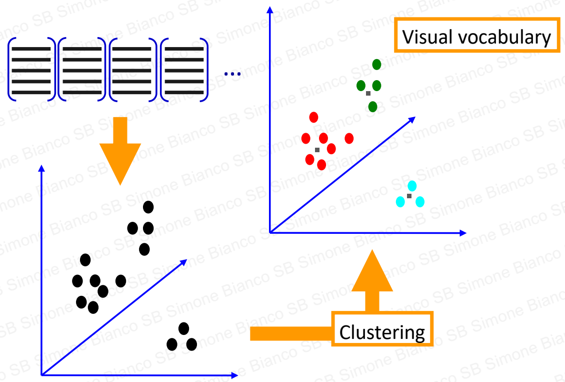
87

Learning the visual vocabulary (2/2)



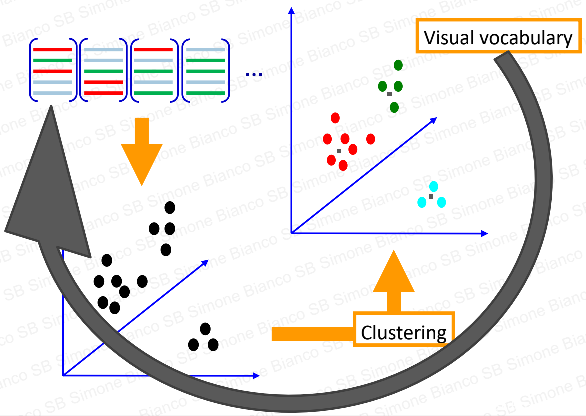
88

Learning the visual vocabulary (2/2)



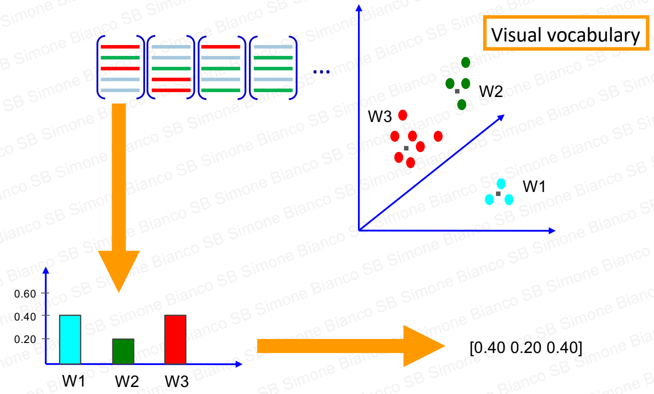
89

BOW encoding



90

BOW encoding

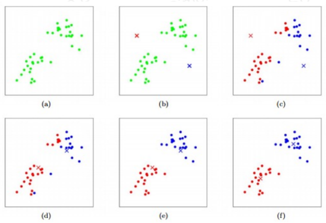


91

Review: K-means clustering

- Want to minimize sum of squared Euclidean distances between features $\mathbf{x}_i$ and their nearest cluster centers $\mathbf{m}_k$

Algorithm: $$D(X,M) = \sum_{\text{cluster } k} \sum_{\substack{\text{point } i \text{ in} \\ \text{cluster } k}} (\mathbf{x}_i - \mathbf{m}_k)^2$$



Example of k-means clustering (k = 2)

- Randomly initialize K cluster centers
- Iterate until convergence:
  - Assign each feature to the nearest center
  - Recompute each cluster center as the mean of all features assigned to it

92

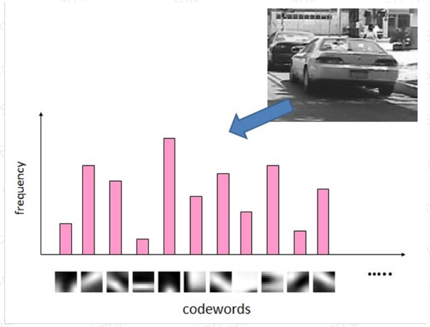
Example visual vocabulary

93

Bag-of-words

Test:

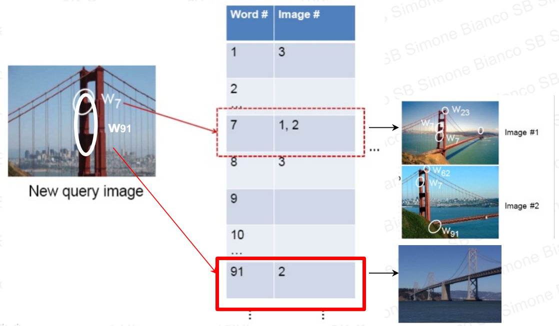
1. Extract descriptors from the image
2. For each descriptor extracted compute its nearest neighbor in the dictionary
3. Build a histogram of length k where the i'th value is the frequency of the i'th dictionary word
4. Classify the histogram with a classifier (e.g. Nearest Neighbor, SVM or Naive Bayes)



94

Applications of Bag of words

Inverted file index and bags of words similarity



95

Bag-of-words

Works with **Deformable Objects**!
Example: tracking

96

Bag-of-words

Works with **Object Classes**!
Example: dogs/parrots classification

97